

	<p style="text-align: center;">Metody programowania 2023/2024 Efektywny QuickSort</p>	<p style="text-align: center;">P_07</p>
---	---	---

Opis

Napisz w Javie program sortujący tablicę $A[0\dots n-1]$ typu long (tablica bez duplikatów) algorytmem QuickSort tak, aby działał w oczekiwanym czasie $O(n\log_2 n)$, nie zawierał wywołań rekurencyjnego i używał pesymistycznie $O(1)$ dodatkowej pamięci (bez implementacji stosu).

Aby uniknąć używania stosu należy zapamiętać położenie prawego końca podtablicy R' w części nieposortowanej tablicy $A[]$ przedstawione na wykładzie. Ponadto algorytm QuickSort należy wykonywać wybierając `pivot` jako „medianę z trzech elementów”, dopóki wielkość podzadań jest większa lub równa 20. Następnie, pozostałe podzadania należy wykonać algorytmem InsertionSort.

Wejście

Dane do programu wczytywane są ze standardowego wejścia (klawiatury) zgodnie z poniższą specyfikacją.

- ❑ Pierwszą podawaną wartością jest dodatnia liczba całkowita oznaczająca ilość zestawów danych, po której na wejściu pojawią się zestawy danych w ilości równej wczytanej liczbie.
- ❑ Każdy zestaw danych zawiera:
 - Dodatnią liczbę całkowitą z zakresu od 1 do 15000 oznaczającą ilość danych wczytywanego zestawu.
 - Zasadnicze dane zestawu w ilości równej poprzednio wczytanej wartości, będące liczbami całkowitymi z przedziału $\langle -2^{63}; 2^{63}-1 \rangle$

Wyjście

Dla każdego zestawu danych, program wypisuje linię zawierającą elementy posortowanej niemalejąco tablicy, oddzielone pojedynczymi odstępami.

Wymagania implementacyjne

Jak w Programie 1, w szczególności kod programu powinien zawierać komentarze szczegółowo opisujące działanie zawartych funkcji oraz testy zgodnie z opisem podanym w Regulaminie zaliczania programów na BaCy.

